SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
BBN REPORT NO. 3479

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
COMMAND AND CONTROL RELATED COMPUTER TECHNOLOGY

5. TYPE OF REPORT & PERIOD COVERED
1 Sept. 1976 - 30 Nov. 1976

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
M. Beeler
J. Burchfiel
R. Nickerson
R. Viswanathan
J. Makhoul
A. W. F. Huggins

8. CONTRACT OR GRANT NUMBER(s)
MDA903-75-C-0180

9. PERFORMING ORGANIZATION NAME AND ADDRESS

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
Bolt Beranek and Newman Inc.
50 Moulton Street, Cambridge, MA 02138

12. REPORT DATE
December 1976

13. NUMBER OF PAGES
48

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
Distribution of this document is unlimited. It may be released to the Clearinghouse, Department of Commerce for sale to the general public.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES
This research was supported by the Defense Advanced Research Projects Agency under ARPA Order No. 2935.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
packet radio, computer communications, PDP-11 TCP, station gateway, ELF, BCPL, cross-radio debugging, speech compression, vocoder, linear prediction, variable data rate compression, speech-quality evaluation.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)
This document describes progress on (1) the development of a packet radio network, (2) speech compression and evaluation. Activities reported under (1) include work on Station Software and Internetworking Research and Development; under (2) development of an automatic variable frame rate scheme for transmitting LPC data, based on our perceptual modeling concept; development of A/D and D/A facility; testing of a new method for reducing sequence effects in subjective assessment of speech quality.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

TABLE OF CONTENTS

I. INTRODUCTION

The packet radio project relies heavily on station software for a variety of control, coordination and monitoring functions. The role of BBN in developing this software is to specify, design, implement and deliver programs which implement these functions.

At the close of the previous quarter we had delivered a working package of station software and documentation to SRI. During this quarter we have improved upon that package and, more important, we have taken the lead in design activities for future packet radio network capabilities. Publication of several design documents has broken new ground in a number of areas. In particular, our route assignment proposal and point-to-point routing proposal are the first and so far the only substantive discussion of these issues in the packet radio context. Our terminal-on-packet and status information papers develop and extend concepts only vaguely addressed heretofore. Other publications concern present development issues and document the present software.

Important progress has been made in areas encompassing and surpassing the packet radio sphere. Transmission Control Protocol has been improved and exported to other ARPA sites. Gateway capability for interconnection of Satellite IMPs and ARPA network IMPs is achieved and ready for delivery in the next quarter.

Noticeable effort has also been expended on continued development and maintenance of software support facilities and hardware. While these account for a minor portion of our work this quarter, the importance of an up-to-date support environment of high quality has received due attention. This and the other areas mentioned above are detailed in the appropriate sections which follow below.

II. MEETINGS, TRIPS AND PUBLICATIONS

This quarter saw two meetings at UCLA and several publications released by BBN. During the week of September 22, UCLA hosted both a meeting on gateways and one on Transmission Control Protocol (TCP). Gateway issues are important to the packet radio project not only to maintain compatibility between packet radio network gateways and other gateways, but also to benefit from design improvements resulting from gateway implementation in other contexts. BBN also has an intense interest in TCP issues, since TCP provides the most practical means to handle internetwork communication. Thus, TCP is in use to permit traffic flow between PRN users and users and facilities on the ARPA network. BBN personnel participated actively in both these meetings. We proposed the version 3 TCP protocol there, as detailed in the TCP section below.

Trips by BBN personnel were occasioned by the two meetings plus gateway delivery. The gateway delivery, to University College of London, will actually be made in the next quarter; one of our group, however, is at UCL as of the end of this quarter. She is involved in final testing and familiarization activities there, preparatory to a meeting and actual delivery.

The documents published by BBN this quarter are:

* PRTN 191 - revision 1, "Terminal-On-Packet Proposal." The original Terminal-On-Packet proposal described a means for end devices, such as terminals, to announce to the station the correspondence between their I.D. and that of their attached PR. At the request of SRI, this traffic is

now being acknowledged by the station when appropriate. Revision 1 of PRTN 191 documents this enhancement.

* PRTN 192, "Route Assignment Proposal."
Suggests a means for choosing routes in the PRN station for assignment to end devices wishing to communicate. Examines the mathematical basis for the specific algorithm ultimately recommended. Object is to provide low delay with adequate reliability. Extends early concepts documented by Ralph Jones (SRI) and Ray Tomlinson (BBN) to a point close to implementation specification. (Note: at close of this quarter, PRTN 192 has been finalized, but distribution will occur slightly into the coming quarter.)

* PRTN 194, "Point-to-Point Routing Proposal."
Tackles the thorny issue of choosing a route between any two packet radio network devices and not necessarily including any specific other device, such as the station. First realistic attack on this crucial design problem. Solution will largely alleviate bottleneck at station caused by present hierarchical routing. No clean solution seems possible, but in PRTN 194 we outline one which is manageable and not particularly contrived. Reaction by Packet Radio Working Group contractors has been mostly favorable, especially so in proportion to the commentator's grasp of the difficulties inherent in the problem. Discussion and resolution of this problem will be an important task for us in the months ahead.

* PRTN 196, "Status Information on SPP Connections."
It has become clear that connections using SPP, the packet radio network protocol for reliable transmission, may benefit greatly from, if not require outright, status information. The ARPA network user is familiar with status messages returned by an IMP or a distant host. Similar connection status messages are proposed and discussed in this document.

* PRTN 199, "Some Station Development Issues."
In our role as designer and implementor of the packet radio station, we at BBN distribute specifications and design discussions in support of our view of the optimal means to achieve ARPA's goals in the packet radio project. This PRTN is such a document. In particular it is a response to PRTN 186, "Station Integration and Design Tasks," in which SRI evidenced a noticeable if not profound lack of realism in their suggestions for station design. PRTN 199 both corrects these misconceptions and replaces them with a positive, realistic design.

* An update to the Packet Radio Station Notebook was prepared and distributed early in this quarter. This fully documents the software transferred to SRI in the summer 1976 delivery.

* XNET, the BBN cross-network debugger (actually now with cross-internetwork capability), has attracted so much attention that a new manual for it has been prepared. This updates descriptions of old commands and documents new ones. It is issued as a stand-alone document, BBN Report No. 3377.

III. TCP

The Internet Transmission Control Program (TCP) has been developed into an operational system and released to SRI for use in Packet Radio Network testing. This resulted in the discovery and subsequent repair of several intricate program bugs which were causing poor performance and occasional garbling of messages.

In order to make the distribution as easy as possible, a "TENEX TCP Installation Guide" was prepared. This is a concise list of all modifications to TENEX which are required for running TCP. It also describes the format of the calls on the TCP as seen from user programs running over it.

During testing it was found that the "zero window problem" manifested itself. This happens when the receiving end of a connection momentarily runs out of buffer space for receiving and tells the sender to stop. Later when space becomes available, that new, non-zero window information must be conveyed to the sender. The original protocol had no reliable way to do this and permitted the connection to stop functioning if the message which announced the window opening was lost in the network. To partially correct this shortcoming, the ARQ (Acknowledge Required) control bit was implemented. This bit uses a sequence number but has no data associated with it; that is, it is something which can be acknowledged and thus guarantee reliable delivery of information (such as the window) which would normally not be acknowledged.

Other protocol changes were made in order to correctly handle half-open connections caused by the host on one end of a connection crashing. Upon restarting, that host may attempt to open the "same" connection and it should not acquire the old half which was left on the host which did not crash -- a new incarnation is required and the old dangling connection must be deleted. The means for achieving this was described at the UCLA TCP meeting and subsequently implemented. The corresponding code for TCPØ which runs in the LSI-11 was also implemented and given to SRI for part of the standard TCPØ.

Because the TCP did not seem as responsive as it should be, extensive testing was done in an attempt to locate the cause. The primary candidate was concerned with lock conflicts. The "standard" implementation of an interprocess interlock involves a test-and-set instruction and a (one second) dismiss and retry loop should the process fail to set the lock. In other words a process is committed to a one-second delay whenever it encounters a lock which has been set by another process. Two other locking mechanisms were tried: one used the TENEX signal-waitfor JSYSs and the other a special lock conflict wait JSYS which was patched in for use by the TCP. Both of these have the characteristic of restarting processes which have conflicted on locks as soon as that lock has been released.

The results were similar with both of these alternate mechanisms. Delay times were reduced significantly but many more

packets were generated. Although some of the causes for this have been identified and cured, there is still a difference between the amount of network traffic with standard locks and with "good" locks.

Another area of suspicion related to excessive delay was the ARPANET itself and the TENEX IMP driver module. A program was written to pass time-stamped messages between two processes on the same host. The operator using this program (SIQTST) could vary the amount of time between the SEND operations. Surprisingly, it was found that the minimum average delay was obtained by having a 20 millisecond delay between the SEND operations! In fact, using zero milliseconds (SENDing as fast as possible) caused the delay to become extremely bad -- typically a second and a half. One reason for this was that TENEX was running out of network buffers; but it appears that the major component of the delay is due to processor bandwidth in the IMP, as there is a significant difference between using a 516 IMP and a 316 IMP which is also supporting TIP functions.

A few short tests were done using type 3 (discardable) messages rather than the normal type 0 (guaranteed delivery) messages. The average delay was approximately 10 percent smaller but 25 percent of the messages were discarded and required retransmission.

The efficiency of the TCP and extensions to the protocol were the subject of a paper presented at the UCLA TCP meeting in

September. This paper described a "Version 3" of the TCP and recommends approximately a dozen changes to the current version 2 protocol. Some of these are compatible with existing TCP implementations and have already been implemented -- such has the half-open connection resolution and the "options" proposal. Currently the header options in use are debugging label (freeing a byte in the TCP header), secure open and secure close. The last two of these are for use by the Network Security Project. The internet timestamp option has been designed but not yet implemented.

Other additions to the TENEX TCP include rewriting the background process and retransmission process in order to minimize the amount of processor time consumed. A second network interface module was written and is being used by the Network Security Project.

Additionally, TCP0 for the PDP-11/40 was modified to fix a bug in the IMP11A driver, use a KW11P clock and a DH11 line driver. A version of TCP0 was written to run in the black side of a BCR device and use the Collins DMA IMP interface. A red side version of TCP0 was also implemented.

IV. GATEWAY

During this quarter we completed work on the software for the Satellite Net/Arpanet gateway. This software was delivered to University College of London (UCL) at the end of the quarter. The software work included:

1) modification of VDH code for incorporation into our ELF system
2) design of a general timestamping mechanism for internetworking applications
3) debugging and checkout of gateway and fake host software

The code required to support the very distant host (VDH) interface consists of a VDH driver and a reliable transmission protocol module (RTP) which implements the VDH protocol as specified in BBN report 1822. Both of these modules had been implemented for an ELF system. The modules interfaced to the ELF Network Control Program, however, a module which is not used in the gateway machine. We modified the VDH driver and RTP modules to interface to the XNCP (Experimental Network Control Program) which handles network communications in the gateway. In the process of modifying these modules to interface to the XNCP, several changes were made to make the software more efficient. The primary change was to allow one process to handle both input and output on the VDH line as opposed to having separate process for input and output. This change eliminated much of the inter-process communication in the previous version.

A method for timestamping packets was proposed by BBN at the UCLA Gateway meeting. Timestamping will be done using the new internet options mechanism. This allows any internet packet to be optionally timestamped by any combination of the SIMPs and gateways through which it passes. Packets can be timestamped at several levels: in the SIMP, in a gateway message generator and in the gateway network interface drivers. These timestamps can be analyzed to measure delays within the gateways, between the gateways and the SIMPs and across the satellite channel. We have coded the timestamp routines in the gateway network interface drivers and will begin debugging these following the delivery of the initial gateway software.

During the quarter, we finished debugging of the gateway and fake host programs for the Satellite Net/Arpanet gateways. The fake host programs include message generators and a statistics gathering facility and will be used to measure delay and bandwidth through the Satellite Net and gateways. Near the end of the quarter, this software was tested using the following configuration. The gateway code was loaded into the PDP-11 gateway machines at UCL and BBN. Message generators and message echoers were alternately run at UCL and BBN, so that messages were generated at either end, then were transmitted to the other gateway which transmitted the messages back to the originating gateway. The messages were transmitted from the UCL PDP-11, via a VDH line to the Goonhilly SIMP. From the Goonhilly SIMP, messages were transmitted back to the London TIP, then via the

NORSAR TIP to the U.S. Within the U.S., messages were transmitted through the Arpanet and across a VDH line to the BBN gateway machine. (Initial testing was done with the configuration explained above. The satellite channel from the Goonhilly SIMP to the ETAM SIMP will be used in all further testing, starting with the delivery of the software to London at the beginning of the next quarter). By the end of the quarter, the gateway and fake host software had been demonstrated by transmitting data between the UCL and BBN gateways and BBN was planning to formally deliver the software to UCL prior to the December 8-10 Gateway and Satellite meetings in London.

V. STATION SOFTWARE DEVELOPMENT AND DELIVERY

V. A. Station Software Delivery

New versions of all the station software were tested at SRI and delivered this quarter, along with updates to the station operator's manual. All testing and delivery were done over the ARPANET from BBN. The changes from the previous station software were as follows.

The new version of ELF (see section VI) was found to reserve some storage in each address space and start the process stack at a higher location than before. Thus all station processes had to be reloaded with a higher start address.

The gateway delivered to SRI last quarter had inadequate buffer space. This restriction was necessary to allow it to fit into SRI's 64K station. Since SRI acquired more station memory this quarter, we were able to increase the buffer space in the gateway.

Last quarter we issued a proposal for Terminal-On-Packets, which would allow the correspondence between end devices and their attached PRs to be automatically determined. As a result of feedback from SRI, a mechanism for getting acknowledgements of TOPs (if desired) was added, and the modified proposal was issued as revision 1 of PRTN 191. The latest control process handles TOPs, ACKing them if appropriate. It also allows the operator to erase end devices from its table by an appropriate command. The

connection process was also upgraded to support these features.

The control process implements an operator dialogue for displaying network connectivity, labeling, etc. This dialogue was originally implemented as part of the main control process, such that the control process could either be talking to the operator or carrying out its normal duties, but not both at once. Once the operator typed a character, the program suspended normal activity and interacted with the operator until he explicitly terminated the dialogue. This is clearly an undesirable situation. The control process was reorganized to run as two ELF processes -- one for the main control functions and one for operator dialogue. The two still cannot be wholly independent, since they must access the same tables. Non-interfering access is implemented using semaphores. The main control process may hang waiting for a semaphore to be released by the dialogue process, but the dialogue process only locks the tables long enough to respond to a single command; it does not hold onto them while awaiting operator input. In conjunction with this reorganization, new commands were added to allow the operator to halt and resume the main control process.

V. B. Continued Station Development

We have continued to develop the station software and anticipate delivery of another version next quarter.

Collins is expected to deliver CAP3, the next version of PR protocol software, early in the next quarter. The only change from CAP2 to CAP3 that affects the connection process and gateway is the addition of a word to packet headers. Both of these programs were modified to accommodate the expanded header and are ready for testing as soon as CAP3 arrives. Changes to the debug and control processes will be more extensive and will have to await receipt of CAP3.

We began implementation of the measurement process this quarter by specifying and implementing collection of label and connectivity data from the control process. This work included:

- Detailed specification of measurement file entries that would be made to record labeling and connectivity;
- Implementation of routines in the control process to create these entries;
- Implementation of operator interaction with the measurement process to specify whether to collect label and/or connectivity data and to turn data collection on and off; and
- Specification and implementation of the interactions between the measurement and control processes to turn on and off data collection and to transfer data.

The operator interaction and measurement-control interaction will extend easily to apply to collection of data from other station processes (e.g., the connection process).

VI. ELF MAINTENANCE AND OTHER SUPPORT SOFTWARE

In September, Packet Radio development work was transferred from a TENEX system on a PDP-10 to a TOPS-20 system on a DECsystem 20. Modifications were made in 11BCPL, XNET and MACN11 in order to run these programs on the new system. In addition, changes were made in MACN11 and in the DECsystem 20 EXEC in order to make MACN11 usable with the DECsystem compile class commands (CCL). As a result of these changes, it is now possible to generate an ELF system from a set of source files using a single CCL command file on the TOPS-20 system. This file controls assembling and linking of all ELF sources and produces an ELF module which can be loaded into the PDP-11 using the XNET debugger.

During this quarter, we brought up a new version of the ELF operating system, released by SRI, for use in the Packet Radio Network station and in the Satellite Net/Arpanet gateways. The primary new feature of this version of the ELF is a faster I/O system. Several errors and undocumented modifications were discovered in this system; these were corrected in our copies of the source files and were reported back to SRI. By the end of the quarter, the new ELF system was performing satisfactorily and we were working with SRI to eliminate a few remaining problems. Another important improvement in our maintenance of the ELF system was the addition of a symbolic debugging capability for the ELF. The ELF is assembled using MACN11 and linked using

LINK11. MACN11 produced symbols for the ELF modules, but LINK11 did not output these symbols in a usable form. Several changes were made to LINK11 to output the symbols provided by MACN11 in a form usable by XNET. The addition of symbols for the ELF system has greater facilitated debugging of that system.

VII. HARDWARE

Three small efforts on packet radio project hardware were executed during this quarter. First, the ANTS interface connecting station PDP-11 number 2 to the ARPA/RCC internetwork suffered performance degradation. The 200 foot cable necessary to reach between the PDP-11 and its internetwork port is significantly longer than the ANTS interface was designed to handle. Careful trimming of cable driving and termination circuits gained temporary use of the interface, only to require repeated, time-consuming adjustment a few hours or days later. The ANTS interface is now considered unusable, and a higher performance replacement has been ordered as discussed below. We are also considering a temporary move of PDP-11 number 2 into physical proximity with its connection. The ANTS interface would probably be usable via the shorter cable then required. This stopgap measure is unattractive, however, due to the wires interconnecting the PDP-11 with neighboring computer bays.

Secondly, one RAM memory board from each PR Digital Unit was sent to Collins for installation of additional thin film packs. This brings the memory complement of each PRDU to the 4 K required to run the new CAP3 software.

The third hardware activity of the quarter involved ordering equipment to upgrade the BBN packet radio stations for measurement functions. This hardware had been itemized in previous quarters; receipt of ARPA funding this quarter has allowed issuing the purchase orders to vendors.

The items involved are:

* Core memory to bring both stations to 128 K.

* Disk, drive, control and spare cartridge for temporary measurement storage. Same equipment as in use at SRI. Can be installed on either PDP-11.

* Line multiplexor (16 lines), line interfaces and cables. Supports ability to print messages and carry on dialogues with experimenter on terminals specific to the task at hand.

* Special interface (IMP11-A) to provide internetwork connection for station number 2. See discussion above.

* Unibus repeater to service added drain imposed by new equipment on station number 2.

Delivery of this hardware, scheduled for completion at mid 1977, will provide a flexible environment for measurement experiments as described in BBN publications.

TABLE OF CONTENTS

I. INTRODUCTION

During the past quarter, we developed an automatic scheme for selecting frames of LPC data for transmission, based on the results and experience gained from our previously completed manual perceptual modeling task [1]. The automatic scheme transmits LPC data at a rate of about 30 frames/sec on the average, out of the available 100 frames/sec analysis data, and it was found to produce a speech quality quite close to that of the speech synthesized with all the analysis frames of data transmitted.

We continued our effort towards developing a real-time interface (A/D and D/A facility) for our PDP-11/SPS-41 system. We checked out the hardware interface to the IMLAC PDS-1 display minicomputer, and wrote and debugged software handlers for this interface.

In speech quality evaluation we tested a newly developed method for reducing sequence effects in subjective assessment of speech quality, by repeating, using this new method, the previously reported pilot study in which we compared subjective quality ratings of variable frame rate systems with the fixed rate systems from which they were derived.

II. DEVELOPMENT OF AN AUTOMATIC PERCEPTUAL MODELING SCHEME

A. Review and Update of our Manual Perceptual Model

The basic assumption underlying our perceptual model is that continuous speech can be represented in terms of LPC parameters extracted at a minimal set of perceptually significant time points, not necessarily equally spaced. In the context of a speech compression system, these time points correspond to instances or frames when transmission of LPC parameters must occur. Thus, the goal of our perceptual modeling task is to develop a perceptually based variable frame rate transmission scheme for LPC parameters, and use it instead of the presently employed likelihood ratio method. We expect that this approach would noticeably enhance the vocoded speech quality.

In our last Quarterly Progress Report [1], we described a manual approach of performing the perceptual modeling task, employing an interactive display program (located on our IMLAC PDS-1 minicomputer) with a capability for playing out selected portions of speech through a D/A converter, and some simple rules dealing only with the information about the transmission parameters: log area ratios (LARs), pitch and gain. Using this approach, transmission mark files, which consist of the positions of the analysis frames for which LARs would be transmitted, were created for six utterances. Excluding the 100 to 200 ms silence segments in the beginning and end of each utterance, the average transmission rate of LARs over the six utterances was found to be

about 30 frames/sec, which represents more than a 3-to-1 reduction from the available 100 frames/sec analysis data. The average number of transmission marks per phoneme came out to be about 2.2. If the above-mentioned silence segments were also taken into account, then the average frame rate dropped to about 27 frames/sec.

It should be pointed out that speech synthesized with LARs transmitted at these low frame rates sounded almost indistinguishable, as judged by informal listening tests, from speech synthesized with all the available 100 frames/sec LARs data transmitted. In these synthesis experiments, we transmitted pitch and gain at the full 100 frames/sec rate, employed a 14-pole LPC analysis, and performed no quantization of any of the transmission parameters.

B. An Automatic Scheme

We developed an automatic scheme for selecting frames of LAR data for transmission, based on the results and experience gained from the manual perceptual modeling scheme reviewed above. The important aspect of the scheme is that, as in the manual procedure, it uses only the information about the transmission parameters. While the developed automatic scheme closely meets our requirements on average transmission frame rate as well as speech quality, we already anticipate several modifications to this scheme. Thus, instead of giving a detailed account of the scheme in its present form, we present below an outline only; the

details will be reported later along with future modifications.

The automatic scheme employs a two-stage procedure, described below, for selecting frames for transmission. In the first stage, a chunk of successive analysis frames of data are considered; the number of frames in the chunk is variable, but its maximum can be specified. In the synthesis experiments discussed later, we chose a maximum of 9 frames. The decision to transmit a frame of data is made in the first stage as follows. Assume that frame n in the current chunk has been marked for transmission, and that frame (n+m) is under consideration. Considering the first LAR, for each of the (m-1) frames that lie between frames n and (n+m), the error between the actual LAR value and the value obtained from linear interpolation between frames n and (n+m) is computed. These (m-1) errors are squared, weighted first by the speech signal energy (in dB) of the corresponding frame and then by a quantity which depends inversely upon the local rate of change of the first LAR, and then finally averaged. This weighted average error is compared against a threshold. If the threshold is exceeded, frame (n+m-1) is marked for transmission; if not, the above procedure is repeated for the second LAR. Currently, the scheme considers up to the fourth LAR only; if the error does not exceed the threshold for all the four LARs, it advances to the frame (n+m+1) and the entire procedure is repeated. Of course, if a frame is marked for transmission, all the LARs will be simultaneously transmitted.

The second stage of the automatic scheme considers the last transmitted frame in the previous chunk and those frames in the present chunk that have been marked for transmission, and attempts to eliminate any unnecessary transmissions. The decision procedure employed in the second stage is the same as in the first stage, except that now the time-averaged error is also averaged over the first four LARs. Our experiments indicated that the second stage deleted about 10% of the transmission marks decided by the first stage.

If, in the first stage, the maximum number of frames in a chunk is chosen as 9, then it is possible to have the maximum interval between successive transmissions to be 17 frames, in view of the action taken by the second stage. This not only means a transmission delay that can be as long as 17 frames or 170 ms, but also the need to store as much as 17 frames of data in the analyzer to be able to compute the error due to the interpolation explained above.

It must be pointed out that the choice of the various parameter values of the weighting functions and the thresholds involved extensive experimentation; we optimized the choice by comparing against the transmission marks that were manually obtained, and by listening to the resulting synthesized speech.

C. Experimental Results and Future Plans

We used the automatic scheme over the same six utterances that we experimented with in our manual perceptual modeling approach. The average frame rate of LAR transmission with the automatic scheme came out to be about 26 frames/sec. Table 1 lists the average frame rates of transmission obtained with the manual and the automatic schemes for each of the six utterances. We also tried the automatic scheme for two other utterances; the frame rates for these are also given in Table 1. (For the listing of the utterances referred to in Table 1, see our Quarterly Progress Report [2].) Notice that the average frame rate depends upon both speech material and speaker. For example, JB1 and JB5 are two different sentences spoken by the same speaker, and their average frame rates are different; on the other hand, RS6, DK6 and DD6 are the same sentence spoken by different speakers, and their average frame rates are also different.

Figure 1 shows the time plots of pitch in Hz (FØ), speech signal energy per sample in decibels (RØ) and the first four LARs (G1, G2, G3, and G4), for the utterance RS6 ("The trouble with swimming is that you can drown", spoken by a female). The long vertical lines mark the frames selected for transmission using our manual perceptual modeling approach. Figs. 2 and 3 are the same as Fig. 1 except that the vertical frame marks were obtained using, respectively, the above automatic scheme and the log

Utterance	Frame Rate of Transmission (fps)	
	Manual Scheme	Automatic Scheme
JB1	22.2	18.1
JB5	30.2	31.5
RS6	26.4	23.4
DK6	25.8	23.4
DD6	29.1	30.8
DK4	28.0	24.3
RS3	24.3	27.6
AR4		35.4
DD2		24.7

Table 1. Average frame rates of LAR transmission for various utterances, for the manual and the automatic perceptual modeling schemes.

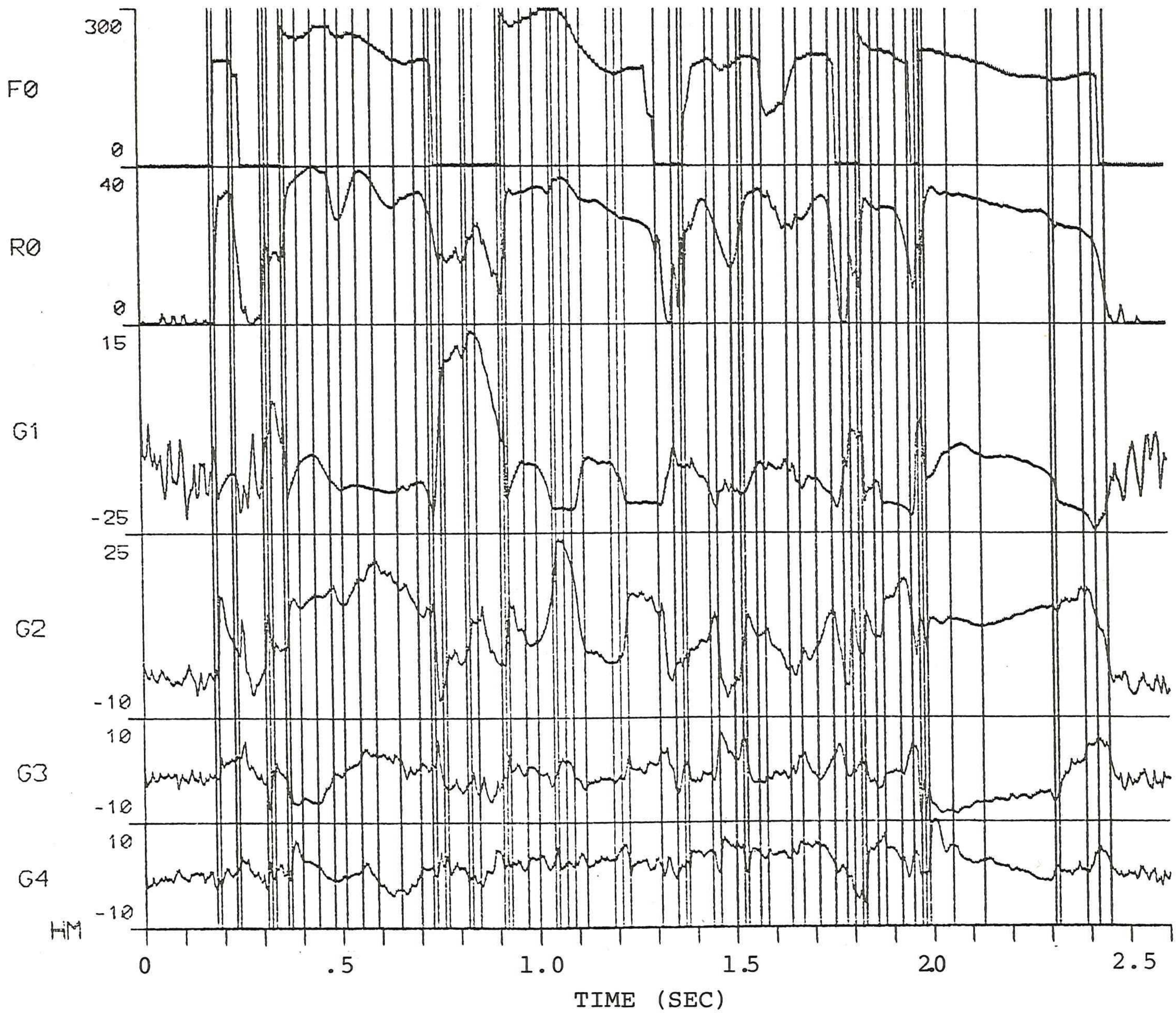

Fig. 1 Time plots of transmission parameters for the utterance RS6, along with transmission marks (long vertical lines) obtained by the manual perceptual modeling scheme.

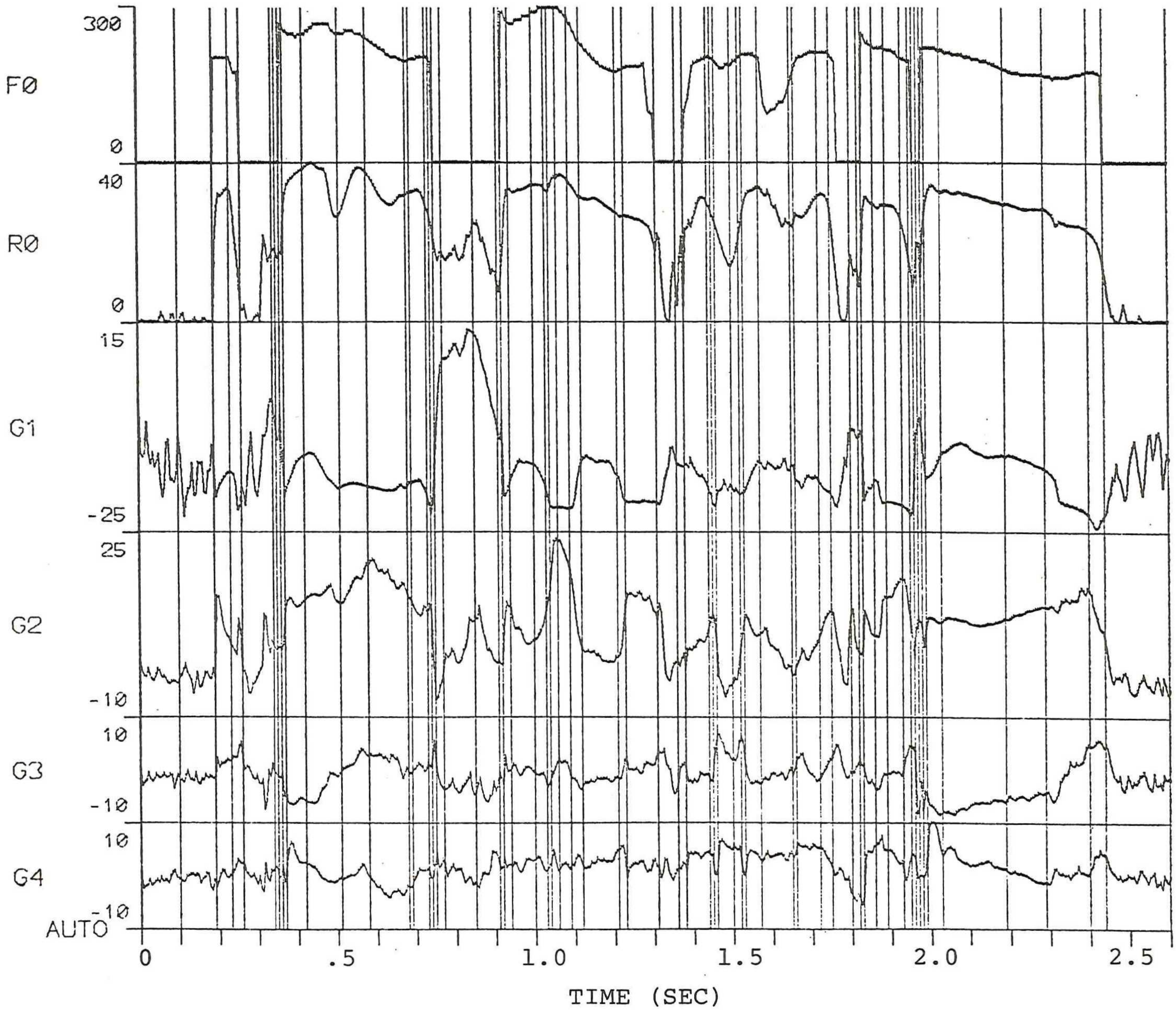

Fig. 2 Time plots of transmission parameters for the utterance RS6, along with transmission marks (long vertical lines) obtained by the automatic perceptual modeling scheme.

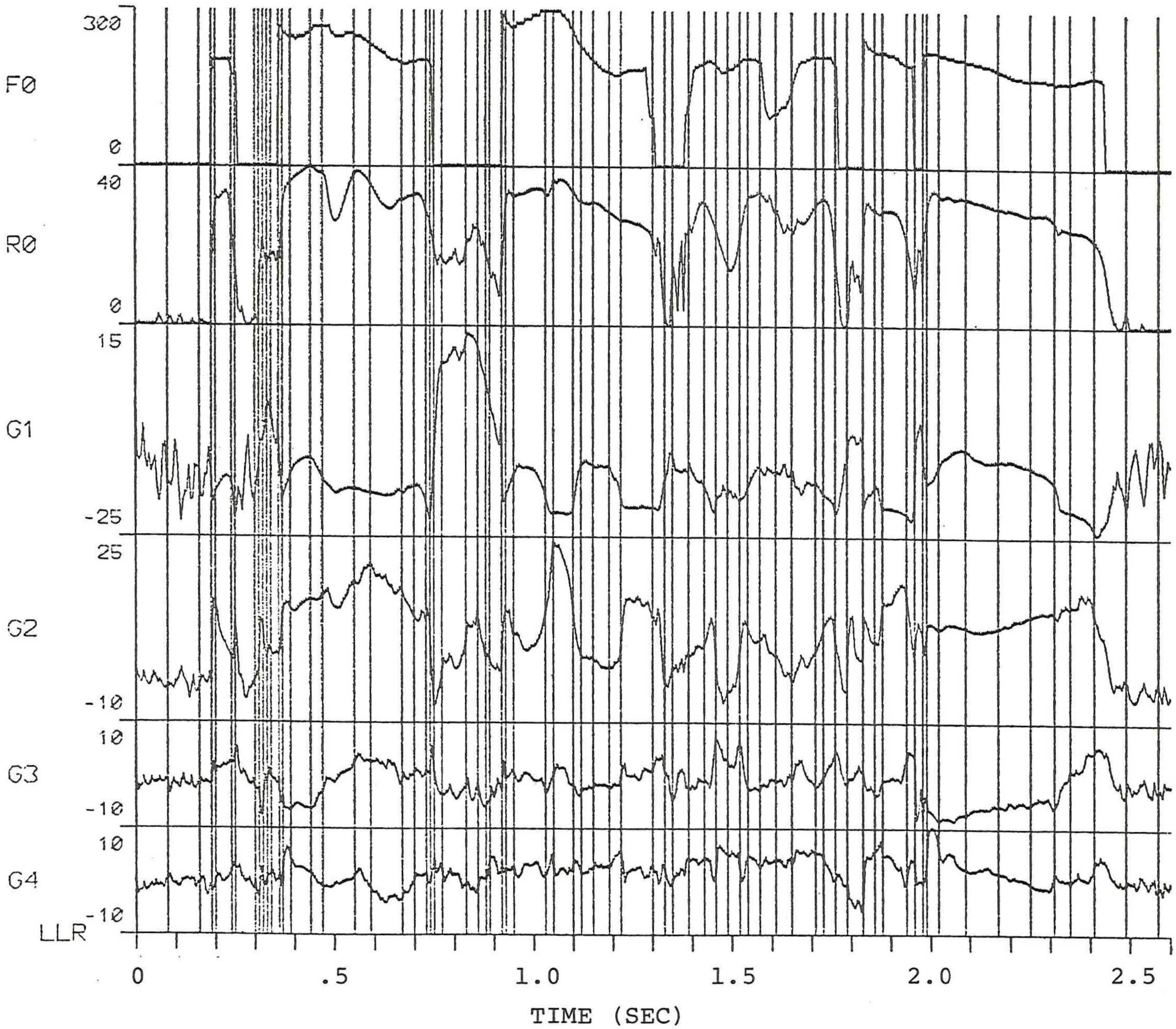

Fig. 3 Time plots of transmission parameters for the utterance RS6, along with transmission marks (long vertical lines) obtained by the log likelihood ratio method.

likelihood ratio method with a single threshold of 1.5 dB [3]. (For RS6, the log likelihood ratio method produced an average transmission frame rate of about 38 frames/sec. Compare this with the corresponding values for RS6 given in Table 1 for the other two methods.)

Informal listening tests conducted on the syntheses obtained from the manual and the automatic perceptual modeling approaches and from the fixed 100 frames/sec system indicated that they all have roughly the same overall quality. An experienced listener could, for some utterances, pick the synthesis from the automatic scheme as being slightly inferior to the syntheses from the other two systems. As before, these syntheses were made with no parameter quantization, with pitch and gain transmitted at 100 frames/sec, and with 14-pole LPC analysis.

Our future plans include modification of the automatic scheme to improve the quality of synthesized speech, and comparative evaluation of the perceptual model with (1) fixed 100 frames/sec system and (2) variable frame rate scheme using log likelihood ratio criterion, under the following conditions: 11-pole (fixed or variable order) LPC analysis, quantization, and variable frame rate transmission of pitch and gain [4]. To this end, we have made an extensive revision of our LPC analysis, encoding, and synthesis programs. The revised programs are more modular which facilitates rapid modifications to any one section of the programs; we can read or write transmission mark files, so

that the syntheses needed for the comparative evaluation task mentioned above can be readily generated.

III. REAL-TIME IMPLEMENTATION

In the last quarter, we continued our effort towards developing a real-time interface (A/D and D/A facility) for our PDP-11/SPS-41 system to replace the one on our PDP-10 which will soon be returned to DEC. We have a working spooling program. Signals can be sampled, stored on the PDP-11 disc, and played back. We also completed checking out the hardware interface to our IMLAC PDS-1 display minicomputer. Software handlers for this interface were written and debugged. Our plans for the future include developing a waveform digitization, display, editing, and playback package using the IMLAC and the above-mentioned A/D and D/A facility. The second phase of FTP (File Transfer Protocol) development, adding the capability to reformat files, is currently being debugged. Its completion will allow us to transfer waveform files to and from TENEX conveniently and quickly.

IV. SPEECH QUALITY EVALUATION

A. A Test of a New Method for Reducing Sequence Effects in Subjective Assessment of Speech Quality

A major difference between tests of intelligibility and those of speech quality is that in the former, there is an objectively correct answer for each test item, whereas in the latter, the responses are judgments for which there is no correct answer. As a result of this, results obtained in present speech quality tests tend to be highly subject to context effects. Thus, the rating assigned by a subject to a particular test item depends not only on the test item itself, but on the range of qualities represented by the other systems under test, and also on which of these other systems were presented for judgment as the preceding two or three stimuli. That is, different ratings are given to a single system, depending on which system was presented on the preceding trial(s).

Sequential effects of this sort have been known for some time. They are a special case of adaptation-level effects described by Helson [5]. The usual method of combatting sequential effects is to counterbalance the presentation sequence, so that every stimulus is preceded equally often by each of the other stimuli in the set. (e.g., [6]). Where large numbers of systems are being compared, this procedure rapidly becomes impractical since the required number of stimulus presentations increases with the square of the number of systems being compared. In the PARM test, developed by Voiers et al [7]

for DCA, the number of stimulus presentations was kept small by comparing only six systems at a time, two of which were anchor systems that appeared in every sextet to provide a baseline for comparing different sextets. However, as Voiers et al point out, even these carefully devised conditions failed to adequately control the sequence effects.

Sequence effects must depend on memory of the perceived quality of the stimuli presented earlier. If the memory could be erased, the sequence effects would disappear. One way of at least weakening memory for an earlier stimulus would be to lengthen the interval between successive stimuli, but this would be incompatible with the aim of minimizing the time taken for testing. A second method is suggested by recent work on auditory short term memory, on the so called suffix effect [8,9]. If a list of items is presented in a short-term memory experiment, and the subject's task is simply to write or repeat the items when the list ends, performance on the last few items is usually relatively error free. But, if an extra item is added to the end of the list (the suffix), performance on the last few items is degraded, even though the subject knows what the suffix will be, and has to make no response to it. The result suggests that the redundant suffix interferes with the auditory memory-trace of the last items in the list. That is, presenting a redundant suffix erases, at least partially, the memory traces of earlier items. Since this is precisely the effect we would like to achieve to reduce sequence effects in quality tests, we have carried out a

pilot study in which we adapted the suffix effect paradigm for this purpose.

Method

The method adopted for erasing auditory memory of the earlier stimuli was to fill the silent intervals between successive stimuli with speech babble. The babble consisted of a carefully controlled mix of six different voices, reading a variety of passages, which was developed as part of a separate BBN project [10]. To test the method, we repeated the earlier pilot study of variable frame rate vocoders reported in [11], using the same subjects. A new stimulus tape was prepared of the same stimuli, in the same presentation order, with the babble faded out and in again one second before and after each stimulus presentation. The fading was controlled by an opto-isolator (Clairex CLM 4006A), mounted on the Language Master in such a way that the card bearing the recorded stimulus interrupted the light from a bulb mounted opposite the isolator. Thus, the babble was faded down whenever a card was put into the Language Master for dubbing onto the stimulus tape, and faded up again whenever the card was removed at the end of the stimulus. The babble was recorded on the second track of the stimulus tape, so that the stimuli could be played either with or without the interleaved babble. The babble was presented at the same level as the signal.

Seven of the eight subjects run in the earlier experiment were still available, so the earlier experiment was repeated, as exactly as possible, except that the inter-stimulus intervals were filled with babble. There were four stimulus sentences (JB5, JB6, RS5, RS6), each processed by eight vocoders, (No. of poles = 12 or 9; LAR quantization step size = 0.5 dB or 2.0 dB; frame rate = 50/sec (fixed) or 23.3/sec (variable)). Thus there were 32 different stimuli. Each was presented four times in a counterbalanced order, with an inter-stimulus interval of six seconds.

Results

The ratings were first sorted by stimulus and subject, retaining the response given to the preceding stimulus. Then, the data were normalized for each subject by subtracting the subject's mean rating from each rating, and dividing by his standard deviation. This was done separately for the responses made with and without babble (i.e., for the results of the earlier pilot, described in [11]). Then the regression line was calculated between (normalized) response, and (normalized) _preceding_ response, separately for each of the 32 stimuli. Since the slope of the regression line is not affected by the mean value of the variables being related, the slope of the lines measure the magnitude of the influence of the preceding response on the present response, independent of which stimulus was presented. The 32 values of slope should have zero mean, if

there is no sequence effect. If the mean is significantly less than zero, a contrastive effect is present, in which the perceived quality of a stimulus is improved if it was preceded by a stimulus with worse quality. If the mean is larger than zero, an assimilative sequence effect is present, in which the perceived quality of a stimulus is decreased when it is preceded by a stimulus with worse quality. The mean slope of the regression lines obtained <u>without</u> babble was +0.210 (S.D. = 0.196), which is significantly different from the expected value of zero ($t = 6.06$, $p<.001$). The mean slope <u>with</u> babble was 0.165 (S.D. = 0.158). This is also significantly larger than zero ($t = 5.91$, $P<.001$). Thus both experiments showed a highly significant assimilative sequence effect. The difference between the mean slopes was not significant, ($t = 1.03$, $P<0.15$, one tailed), although the difference is in the right direction to suggest that the babble may have reduced the sequence effect slightly. In support of this, all the subjects said they found it harder to compare a stimulus with its predecessor when the babble was present, which suggests that the ratings would be less influenced by sequence effects.

Since the results show that there was no significant difference in sequence effects as a result of introducing the babble, the two experiments can be treated as replications, to check on the reliability of our method. Plots of the data with and without babble show that both experiments yielded highly similar results, except that the absolute level of ratings

assigned with babble was slightly lower without the babble, perhaps because the babble consisted of a mixture of voices recorded under good conditions, and thus may have acted as an undegraded anchor against which the eight vocoder systems appeared more degraded than in the absence of the babble.

B. Analysis of the Subjective Rating Study of the Effects on Quality of Varying Vocoder Parameters

The analysis of this study (see [1]) is proceeding, but we will postpone presentation of the results until all analyses are complete. The data have already been subjected to analysis of variance, and to multi-dimensional scaling with MDPREF. Unfortunately, with 50 different vocoders to represent within the analysis space, it is very difficult to discover the rotation of the solution that is easiest to interpret. We have recently learned that new developments [12] have made it possible to apply INDSCAL to rating data, which avoids the rotation problem since INDSCAL produces a unique rather than a rotatable solution. We have been in communication with Bell Labs, and hope to apply INDSCAL to our data--and perhaps, to earlier data--in the near future.

REFERENCES

1. BBN Quarterly Progress Report, Command and Control Related Computer Technology, BBN Report No. 3430, September 1976.

2. BBN Quarterly Progress Report, Command and Control Related Computer Technology, BBN Report No. 3093, June 1975.

3. R. Viswanathan and J. Makhoul, "Specifications for ARPA-LPC System II," NSC Note 82, Feb. 1976.

4. R. Viswanathan, "Variable Frame Rate Transmission of Pitch and Gain," NSC Note 96, September 1976.

5. H. Helson, "Adaptation-Level as a Basis for a Quantitative Theory of Frames of Reference," Psych. Review, Vol. 55, pp. 297-313, 1948.

6. A.W.F. Huggins, "The Perception of Timing in Natural Speech I: Compensation Within the Syllable," Language and Speech, Vol. 11, pp. 1-11, 1968.

7. W.D. Voiers et al, "Narrowband Digital Voice Processor Consortium Evaluation Methods," Appendix B of the Consortium Report Draft (made available unofficially to BBN).

8. R.G. Crowder and J. Morton, "Precategorical Acoustic Storage (PAS)," Perception and Psychophysics, Vol. 5, pp. 365-373, 1969.

9. R.G. Crowder, "Audition and Speech Coding in Short-Term Memory: a Tutorial Review," presented at Attention and Performance VII, Senanque, France, August 1976.

10. D.N. Kalikow, K.N. Stevens, and L.L. Elliott, "Development of a Test of Speech Intelligibility in Noise Using Sentence Materials with Controlled Word Predictability," BBN Report No. 3370, submitted for publication, Journal of the Acoustical Society of America, 1976.

11. BBN Quarterly Progress Report, Command and Control Related Computer Technology, BBN Report NO. 3325, June 1976.

12. J.B. Kruskal, "More Factors than Subjects, Tests, and Treatments: An Indeterminacy Theorem for Canonical Decomposition and Individual Differences Scaling," Psychometrica, Vol. 41, pp. 281-293, 1976.